In [*]:
```
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt

%matplotlib inline
```

In [ ]:
```
df=pd.read_csv("Housing.csv")
```

In [ ]:
```
df
```

In [ ]:
```
# to find number of rows and columns
df.shape
```

In [ ]:
```
#general info
df.info()
```

In [ ]:
```
#show top 5 rows
df.head()
```

```
In [3]: df
```

Out[3]:

	longitude	latitude	housing_median_age	total_rooms	total_bedrooms	population	households	median_income	me
0	-122.23	37.88	41.0	880.0	129.0	322.0	126.0	8.3252	
1	-122.22	37.86	21.0	7099.0	1106.0	2401.0	1138.0	8.3014	
2	-122.24	37.85	52.0	1467.0	190.0	496.0	177.0	7.2574	
3	-122.25	37.85	52.0	1274.0	235.0	558.0	219.0	5.6431	
4	-122.25	37.85	52.0	1627.0	280.0	565.0	259.0	3.8462	
...	...	...	...	...	...	...	...	...	
20635	-121.09	39.48	25.0	1665.0	374.0	845.0	330.0	1.5603	
20636	-121.21	39.49	18.0	697.0	150.0	356.0	114.0	2.5568	
20637	-121.22	39.43	17.0	2254.0	485.0	1007.0	433.0	1.7000	
20638	-121.32	39.43	18.0	1860.0	409.0	741.0	349.0	1.8672	
20639	-121.24	39.37	16.0	2785.0	616.0	1387.0	530.0	2.3886	

20640 rows × 10 columns

```
 6   households          20640 non-null  float64
 7   median_income       20640 non-null  float64
 8   median_house_value  20640 non-null  float64
 9   ocean_proximity     20640 non-null  object
dtypes: float64(9), object(1)
memory usage: 1.6+ MB
```

In [ ]:
```
#show top 5 rows
df.head()
```

In [ ]:
```
#extract all columns of dataset
df.columns
```

In [ ]:
```
#check for all null values
df.isna().sum()
```

In [ ]:
```
df.dropna(inplace=True)
```

In [ ]:
```
#check for all null values
df.isna().sum()
```

In [ ]:
```
df.describe()
```

```
total_bedrooms        207
population              0
households              0
median_income           0
median_house_value      0
ocean_proximity         0
dtype: int64
```

In [ ]:
```
df.dropna(inplace=True)
```

In [ ]:
```
#check for all null values
df.isna().sum()
```

In [ ]:
```
df.describe()
```

In [ ]:
```
sns.pairplot(df)
```

In [ ]:
```
#Label encoding and assign in new variable
from sklearn import preprocessing
Label_encode = preprocessing.LabelEncoder()
```

In [ ]:
```
#Assign in new variable
```

```
df.isna().sum()
```

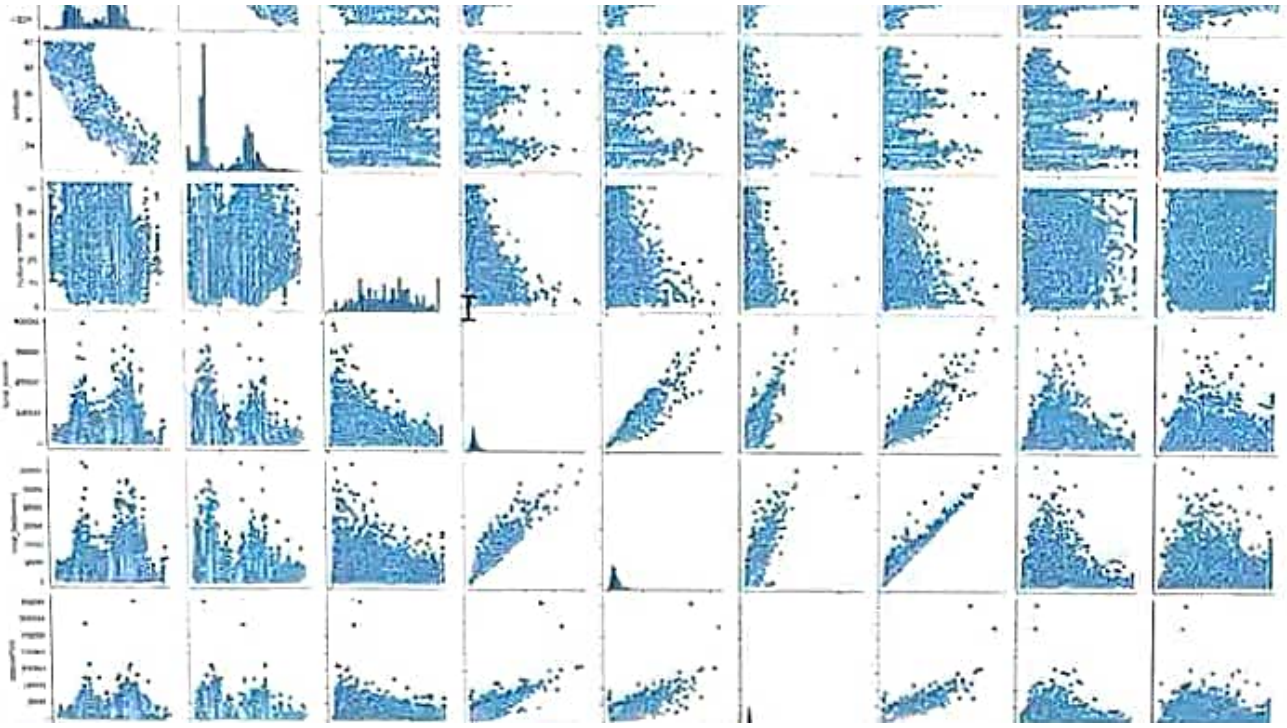
Out[10]:
```
longitude             0
latitude              0
housing_median_age    0
total_rooms           0
total_bedrooms        0
population            0
households            0
median_income         0
median_house_value    0
ocean_proximity       0
dtype: int64
```

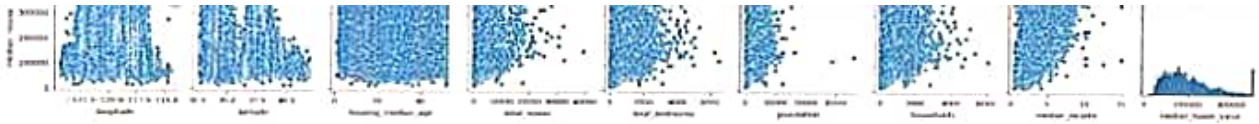
In [ ]:
```
df.describe()
```

In [ ]:
```
sns.pairplot(df)
```

In [ ]:
```
#Label encoding and assign in new variable
from sklearn import preprocessing
Label_encode = preprocessing.LabelEncoder()
```

In [ ]:
```
#Assign in new variable
df['oceanproximity']=Label_encode.fit_transform(df['ocean_proximity'].values)
```





In [13]:
```
#Label encoding and assign in new variable
from sklearn import preprocessing
Label_encode = preprocessing.LabelEncoder()
```

In [ ]:
```
#Assign in new variable
df['oceanproximity']=Label_encode.fit_transform(df['ocean_proximity'].values)
```

In [ ]:
```
#check assigned values
m=df.groupby('ocean_proximity')
m=m['oceanproximity']
m.first()
```

In [ ]:
```
sns.heatmap(df.corr(),annot=True)
```

In [ ]:
```
df.corr()
```

In [ ]:
```
#feature selection
```

Out[15]:
```
ocean_proximity
<1H OCEAN     0
INLAND        1
ISLAND        2
NEAR BAY      3
NEAR OCEAN    4
Name: oceanproximity, dtype: int32
```

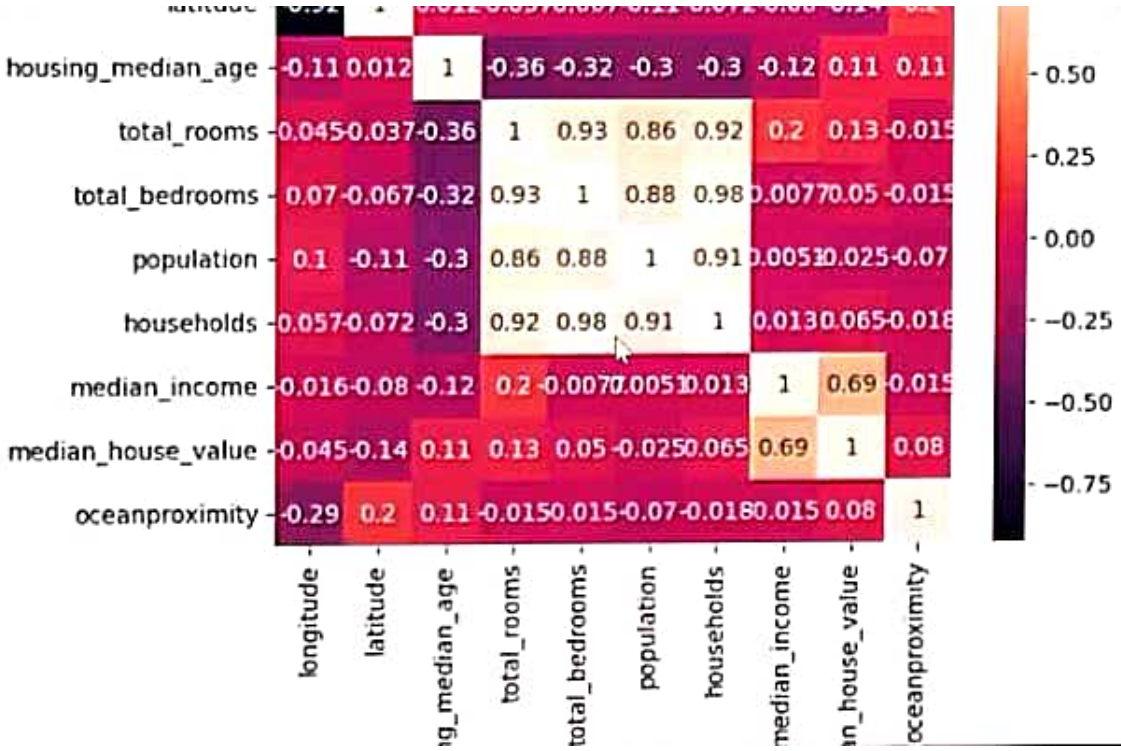
In [ ]:
```
sns.heatmap(df.corr(),annot=True)
```

In [ ]:
```
df.corr()
```

In [ ]:
```
#feature selection
columns=['longitude','latitude','housing_median_age','total_rooms','total_bedrooms','population','h
x=df[columns]
y=df['median_house_value']
```

In [ ]:
```
print(x)
```

In [ ]:
```
print(y)
```


housing_median_age
total_rooms
total_bedrooms
population
households
median_income
median_house_value
oceanproximity
-0.11 0.012 1 -0.36 -0.32 -0.3 -0.3 -0.12 0.11 0.11
-0.045 -0.037 -0.36 1 0.93 0.86 0.92 0.2 0.13 -0.01
0.07 -0.067 -0.32 0.93 1 0.88 0.98 0.007 0.05 -0.01
0.1 -0.11 -0.3 0.86 0.88 1 0.91 0.005 0.025 -0.07
-0.057 -0.072 -0.3 0.92 0.98 0.91 1 0.013 0.065 -0.01
-0.016 -0.08 -0.12 0.2 -0.007 0.005 0.013 1 0.69 -0.01
-0.045 -0.14 0.11 0.13 0.05 -0.025 0.065 0.69 1 0.08
-0.29 0.2 0.11 -0.015 0.015 -0.07 -0.018 0.015 0.08 1
longitude
latitude
total_rooms
total_bedrooms
population
households
median_income
oceanproximity
0.50
0.25
0.00
−0.25
−0.50
−0.75

housing_ tot me median_ oc

In [ ]:
```
df.corr()
```

In [ ]:
```
#feature selection
columns=['longitude','latitude','housing_median_age','total_rooms','total_bedrooms','population','h
x=df[columns]
y=df['median_house_value']
```

In [ ]:
```
print(x)
```

In [ ]:
```
print(y)
```

In [ ]:
```
from sklearn.model_selection import train_test_split
X_train, X_test, Y_train, Y_test = train_test_split(
    x, y, train_size=0.7, test_size=0.3)
```

In [ ]:
```
from sklearn.linear_model import LinearRegression
```

population	0.100270	-0.108997	-0.295787	0.857281	0.877747	1.000000	0.907186
households	0.056513	-0.071774	-0.302768	0.918992	0.979728	0.907186	1.000000
median_income	-0.015550	-0.079626	-0.118278	0.197882	-0.007723	0.005087	0.013434
median_house_value	-0.045398	-0.144638	0.106432	0.133294	0.049686	-0.025300	0.064894
oceanproximity	-0.289530	0.200801	0.112330	-0.015363	-0.014768	-0.069630	-0.018251

In [ ]:
```
#feature selection
columns=['longitude','latitude','housing_median_age','total_rooms','total_bedrooms','population','h
x=df[columns]
y=df['median_house_value']
```

In [ ]:
```
print(x)
```

In [ ]:
```
print(y)
```

In [ ]:
```
from sklearn.model_selection import train_test_split
X_train, X_test, Y_train, Y_test = train_test_split(
    x, y, train_size=0.7, test_size=0.3)
```

```
x=df[columns]
y=df['median_house_value']
```

In [19]:
```
print(x)
```

```
       longitude  latitude  housing_median_age  total_rooms  total_bedrooms  \
0        -122.23     37.88                41.0        880.0           129.0
1        -122.22     37.86                21.0       7099.0          1106.0
2        -122.24     37.85                52.0       1467.0           190.0
3        -122.25     37.85                52.0       1274.0           235.0
4        -122.25     37.85                52.0       1627.0           280.0
...          ...       ...                 ...          ...             ...
20635    -121.09     39.48                25.0       1665.0           374.0
20636    -121.21     39.49                18.0        697.0           150.0
20637    -121.22     39.43                17.0       2254.0           485.0
20638    -121.32     39.43                18.0       1860.0           409.0
20639    -121.24     39.37                16.0       2785.0           616.0

       population  households  median_income  oceanproximity
0           322.0       126.0         8.3252               3
1          2401.0      1138.0         8.3014               3
2           496.0       177.0         7.2574               3
3           558.0       219.0         5.6431               3
4           565.0       259.0         3.8462               3
```

```
20638       741.0       349.0         1.8672              1
20639      1387.0       530.0         2.3886              1

[20433 rows x 9 columns]
```

In [ ]:
```
print(y)
```

In [ ]:
```
from sklearn.model_selection import train_test_split
X_train, X_test, Y_train, Y_test = train_test_split(
    x, y, train_size=0.7, test_size=0.3)
```

In [ ]:
```
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_absolute_error
model = LinearRegression()
model.fit(X_train, Y_train)
Y_pred = model.predict(X_test)

model.score(X_test,Y_test)
```

In [ ]:
```
from sklearn.metrics import r2_score
score = r2_score(Y_test,Y_pred)
print("The accuracy of our model is {}%".format(round(score, 2) *100))
```